LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-SM-212168

Interface Documentation for MS Material Models

R. Becker

May 11, 2005

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Revised interface for MS material models

The interface for the MS material model has been modified to utilize results from EOS calculations performed in the host code. This is thought to be feasible for the following models:

- All of the hardening models associated with the traditional von Mises yield surface (J2-Flow theory) including most of the loosely coupled fractured models. This includes the Moss-Attia-Rubin (marfract) model.
- Simple anisotrpoic materials, but this may prove to be more general
- The Gurson model with the assumption that the bulk modulus is constant over the range of pressures at which voids can exist. This is not a bad assumption considering that voids vanish at high pressure.
- Crystal plasticity for cubic crystal structures and possibly others with sufficient approximations.

Such decoupling will not be possible for other models in the MS suite because of very tight coupling between pressure and the deviatoric stress:

- DTRA concrete model
- Cam Clay soil model
- Models imported through an ABAQUS / UMAT interface
    - BCJ fracture model
    - VISCOSCRAM fracture
- Geodyn material models

Modification to the MS models will be staged. The first round, which is complete, includes the J2-Flow theory models where code modifications have been minimal. We should be able to work out bugs and address other issues with this set of models. The Gurson model will need to be rewritten to replace the deeply rooted calls to the EOS routines with a calculation using the trial pressure and a bulk modulus.

The table below lists three interfaces that are supported in the initial implementation and what variables are utilized. The ALE3D call will use the EOS from the MS models, including the energy update and bulk viscosity. Two other interfaces are implemented. One is similar to the ALE3D call using the EOS models in the MS suite, “EOS from MS”. The QdV work from the energy is removed, and the effects of dissipative work on the pressure are removed using dp/de.

The final option is represented by “EOS: from host”. Here the host code computes the pressure and updates the energy for PdV and QdV work. The MS models modify the energy and pressure given for the strain work dt/de and dp/de. For material with volume damage and the Gurson model, the pressure, PdV work and temperature will be adjusted to account for the plastic/crack-induced volume change. For this, the bulk modulus, dpde and dTde are needed from the host EOS package. One special factor that the host code requires is a measure of volume expansion due to damage. This can take the form of a ratio of solid density to zonal density (always > 1), which can be used as a multiplicative factor on the density passed to the EOS routine. This will prevent large negative pressures in failed zones.

			EOS: from host			EOS: from MS			ALE3D	
VARIABLE	**WHAT IT IS**	**DESCRIPTION**	**IN**	**OUT**	**S**	**IN**	**OUT**	**S**	**IN**	**OUT**
vol_ratio[0]	V/V_0 (ρ_0/ρ) at t	Relative volume at t	x			x			x	
vol_ratio[1]	V/V_0 (ρ_0/ρ) at t+Δt	Relative volume at t+Δt	x			x			x	
vol_ratio[2]	$\dot{V}/V$	Volume strain rate	x			x			x	
vol_ratio[3]	$\Delta V/V_0$	Normalized volume increment	x			x			x	
def_rate[0]	$\dot{\varepsilon}'_{xx} = \dot{u}/dx - \frac{1}{3}\dot{V}/V$	Deviatoric part of xx strain rate	x			x			x	
def_rate[1]	$\dot{\varepsilon}'_{yy} = \dot{v}/dy - \frac{1}{3}\dot{V}/V$	Deviatoric part of yy strain rate	x			x			x	
def_rate[2]	$\dot{\varepsilon}'_{zz} = \dot{w}/dz - \frac{1}{3}\dot{V}/V$	Deviatoric part of zz strain rate	x			x			x	
def_rate[3]	$\dot{\varepsilon}_{yz} = \frac{1}{2}\left(\dot{w}/dy + \dot{v}/dz\right)$	Deviatoric part of yz strain rate	x			x			x	
def_rate[4]	$\dot{\varepsilon}_{zx} = \frac{1}{2}\left(\dot{u}/dz + \dot{w}/dx\right)$	Deviatoric part of zx strain rate	x			x			x	
def_rate[5]	$\dot{\varepsilon}_{xy} = \frac{1}{2}\left(\dot{v}/dx + \dot{u}/dy\right)$	Deviatoric part of xy strain rate	x			x			x	
def_rate[6]	$\dot{V}/V$	Volume strain rate	x			x			x	
spin[0]	$w_x = -\frac{1}{2}\left(\dot{v}/dz - \dot{w}/dy\right)$	Spin about x	x			x			x	
spin[1]	$w_y = -\frac{1}{2}\left(\dot{w}/dx - \dot{u}/dz\right)$	Spin about y	x			x			x	
spin[2]	$w_z = -\frac{1}{2}\left(\dot{u}/dy - \dot{v}/dx\right)$	Spin about z	x			x			x	
rho0	ρ_0	Initial density	x			x			x	
deltaTime	Δt	Time increment	x			x			x	
ss	C_l	Longitudinal sound speed					x	x		x
stress[0]	$\sigma'_{xx} = \sigma_{xx} + p$	Deviatoric stress component xx	x			x			x	
stress[1]	$\sigma'_{yy} = \sigma_{yy} + p$	Deviatoric stress component yy	x			x			x	
stress[2]	$\sigma'_{zz} = \sigma_{zz} + p$	Deviatoric stress component zz	x			x			x	
stress[3]	σ_{yz}	Stress component yz	x			x			x	
stress[4]	σ_{zx}	Stress component zx	x			x			x	
stress[5]	σ_{xy}	Stress component xy	x			x			x	
stress[6]	p	Pressure	x			x			x	

energy[0]	e	Energy per unit ref. vol at t	x			x			x	
energy[1]										
energy[2]										
energy[3]										
energy[4]										
energy[5]										
energy[6]										
energy[7]										
tempk	T	Temperature in K	x							
stress_updt[0]	$\sigma'_{xx} = \sigma_{xx} + p$	Deviatoric stress component xx		x			x			x
stress_updt[1]	$\sigma'_{yy} = \sigma_{yy} + p$	Deviatoric stress component yy		x			x			x
stress_updt[2]	$\sigma'_{zz} = \sigma_{zz} + p$	Deviatoric stress component zz		x			x			x
stress_updt[3]	σ_{yz}	Stress component yz		x			x			x
stress_updt[4]	σ_{zx}	Stress component zx		x			x			x
stress_updt[5]	σ_{xy}	Stress component xy		x			x			x
stress_updt[6]	p	Pressure	x	x			x			x
energy_updt[0]	e	Energy per unit ref. vol. at t+Δt	x	x	x		x			x
energy_updt[1]	e_c	Cold energy per unit ref. volume								x
energy_updt[2]		QdV work								x
energy_updt[3]		Reversible volume work								x
energy_updt[4]		Plastic volume work								x
energy_updt[5]		Total volume work								x
energy_updt[6]		Reversible shear work								x
energy_updt[7]		Irreversible shear work								x
tempk_updt		Temperature at t+Δt		x			x			x
ysModel	enum type	Yield surface model	x						x	
ys_const[n]	params	Yield surface constants	x			x			x	
ys_hist[n]	state vars	State variables code maintains	x			x			x	
ys_hist_updt[n]	state vars	Updated state variables		x			x			x
ys_hist_updt[x]	$V_f = \rho_0 / \rho_Z$	Volume change from damage		x	x					

eosModel	enum type	Eos model	x			x			x	
eos_const[n]	params	Eos surface constants				x			x	
eos_hist[n]	state vars	State variables code maintains				x			x	
eos_hist_updt[n]	state vars	Updated state variables					x			x
eos_hist_updt[x]	K	Bulk modulus	x		x					
elasModel	enum type	Elastic shear model	x			x			x	
elas_const[n]	params	Eos surface constants	x			x			x	
elas_hist[n]	state vars	State variables code maintains	x			x			x	
elas_hist_updt[n]	state vars	Updated state variables		x			x			x
hardModel	enum type	Elastic shear model	x			x			x	
hard_const[n]	params	Eos surface constants	x			x			x	
hard_hist[n]	state vars	State variables code maintains	x			x			x	
hard_hist_updt[n]	state vars	Updated state variables		x			x			x
failModel	enum type	Elastic shear model	x			x			x	
fail_const[n]	params	Eos surface constants	x			x			x	
fail_hist[n]	state vars	State variables code maintains	x			x			x	
fail_hist_updt[n]	state vars	Updated state variables		x			x			x
melted	int	SOLID=0 or MELTED=1	x		x					
bulkQ[0]	Q flag	For Q calc: turn on and off				x			x	
bulkQ[1]	linear Q	For Q calc: linear Q factor				x			x	
bulkQ[2]	quadratic Q	For Q calc: quadratic Q factor				x			x	
bulkQ_updt	Q	Q from volume strain rate					x	x		x
dpdv	dp/dV	Pressure change with volume					x			x
dpde	dp/de	Gruneissen coefficient	x				x			x
dtde	dT/de	Specific heat x density	x			x			x	
msModel	ε vs σ	Tabular stress strain curve			x			x	x	
rot_method	enum type	Stress rotation (Jaumann/none)	x		x	x		x	x	
stress_operation	enum type	Deformation step or remap	x		x	x		x	x	
ddsdde[6][6]	**K**	Material stiffness for implicit								x
ddsdt[6]	**M**	Thermal stress dependence: impl.								x

IN-> input to MS model; OUT-> output from MS model; S-> variables requiring special consideration.

Host code supplies EOS. Material routines modify pressure and temperature for effects of dissipated energy

```
void MS_MaterialHostEOS (
         MS_Matmodel_t            msModel,
         DeformOrRemap_t          stress_operation,
         StressRateObjectivity_t  rot_method,
         real8 *vol_ratio,     real8 *def_rate,      real8 *spin,
         real8 rho0,           real8 deltaTime,      int   melted,
         real8 *ss,            real8 bulkMod,        real8 *v_damage,
         real8 *stress,        real8 *energy,        real8 tempk,
         real8 *stress_updt,   real8 *energy_updt,   real8 *tempk_updt,
         MS_Model_t ysModel,   real8 *ys_params,     real8 *ys_hist,
                               int   *ys_opts,       real8 *ys_hist_updt,
         MS_Model_t elasModel, real8 *elas_params,   real8 *elas_hist,
                               int   *elas_opts,     real8 *elas_hist_updt,
         MS_Model_t hardModel, real8 *hard_params,   real8 *hard_hist,
                                                     real8 *hard_hist_updt,
         MS_Model_t failModel, real8 *fail_params,   real8 *fail_hist,
                                                     real8 *fail_hist_updt,
         real8 dpde_n,         real8 dtde_n )
```

Standard ALE3D interface where material model supplies all material response plus bulk Q

```
void MS_MaterialModels (
         MS_Matmodel_t            msModel,
         DeformOrRemap_t          stress_operation,
         StressRateObjectivity_t  rot_method,
         real8 *vol_ratio,     real8 *def_rate,      real8 *spin,
         real8 rho0,           real8 deltaTime,      int   melted,
         real8 *ss,            real8 bulkMod,        real8 *v_damage,
         real8 *stress,        real8 *energy,        real8 tempk,
         real8 *stress_updt,   real8 *energy_updt,   real8 *tempk_updt,
         real8 *bulkQ,         real8 *bulkQ_updt,
         MS_Model_t ysModel,   real8 *ys_params,     real8 *ys_hist,
                               int   *ys_opts,       real8 *ys_hist_updt,
         MS_Model_t eosModel,  real8 *eos_params,    real8 *eos_hist,
                                                     real8 *eos_hist_updt,
         MS_Model_t elasModel, real8 *elas_params,   real8 *elas_hist,
                               int   *elas_opts,     real8 *elas_hist_updt,
         MS_Model_t hardModel, real8 *hard_params,   real8 *hard_hist,
                                                     real8 *hard_hist_updt,
         MS_Model_t failModel, real8 *fail_params,   real8 *fail_hist,
                                                     real8 *fail_hist_updt,
         int *aniso_opts,      real8 *aniso_params,  real8 *aniso_hist,
                                                     real8 *aniso_hist_updt,
         real8 *dpdv,          real8 *dpde,          real8 *dtde,
         real8 ddsdde[6][6],   real8 ddsdt[6],       real8 dtdde[6],
         MatFlags_t matFlags,  MatFlags_t retFlags)
```

Material routine calculates deviatoric stress and pressure. Bulk viscosity effects are removed from PdV work

```
void MS_MaterialHostQ (
         MS_Matmodel_t            msModel,
         DeformOrRemap_t          stress_operation,
         StressRateObjectivity_t  rot_method,
         real8 *vol_ratio,     real8 *def_rate,      real8 *spin,
         real8 rho0,           real8 deltaTime,      real8 *ss,
         real8 *stress,        real8 *energy,        real8 tempk,
         real8 *stress_updt,   real8 *energy_updt,   real8 *tempk_updt,
         MS_Model_t ysModel,   real8 *ys_params,     real8 *ys_hist,
                               int   *ys_opts,       real8 *ys_hist_updt,
         MS_Model_t eosModel,  real8 *eos_params,    real8 *eos_hist,
                                                     real8 *eos_hist_updt,
         MS_Model_t elasModel, real8 *elas_params,   real8 *elas_hist,
                               int   *elas_opts,     real8 *elas_hist_updt,
         MS_Model_t hardModel, real8 *hard_params,   real8 *hard_hist,
                                                     real8 *hard_hist_updt,
         MS_Model_t failModel, real8 *fail_params,   real8 *fail_hist,
                                                     real8 *fail_hist_updt,
         real8 *dpdv,          real8 *dpde,          real8 *dtde)
```